

# Contention Management: An overview

---

*Predictable Network Solutions Ltd*

9 August 2011

## Overview

Imagine a packet network carrying statistically-multiplexed traffic of multiple types (e.g. real-time voice, streaming video and bulk data). Each application has a set of desired performance outcomes, such as voice quality, or Web advert load time. These outcomes are dependent on network end-to-end loss and delay, which we call *quality attenuation*. The basic premise of our technology is that a necessary precondition for these outcomes to be *assured* is that the delivered quality attenuation is within acceptable bounds. The majority of that quality attenuation accrues at just a few points along the path. By shifting that quality attenuation to a *single* control point at ingress to the network, the loss and delay budget can be dynamically traded between streams. This creates a qualitative step-change in the cost structure and value that packet networks can offer.

## Today's approach to contention: control loops

Data networks (like any statistically multiplexed routed resource) can get into trouble much faster than they can get out. They are in 'trouble' when they experience the effects of excessive short-term loading, namely excessive loss, delay and the delay variation. They can only recover in a linear fashion, while they can get into trouble algebraically (if not exponentially) fast.

There are multiple timescales over which data networks have to be managed to avoid 'trouble' – trying to dynamically match demand to supply. The current focus is to apply control loops (like TCP) to infer downstream network conditions and adjust to offered load. Such approaches may be valid over multiple periods of round-trip times (e.g.  $10^0$  or  $10^1$  seconds and above) but below those timescales they are not effective (as elementary control theory would tell us). This was much less of an issue when use of networks was bulk data, in which packets are relatively time-insensitive. It becomes a great problem when we multiplex in time-sensitive traffic as we need to over-provision capacity (at a very high cost) to reduce the overall quality attenuation – and even over-provisioning has limits on its effectiveness as traffic goes from a 'fast' to a 'slow' network (e.g. FTTH to a slower WLAN).

Delay can build very quickly in switching elements<sup>1</sup>. These systems can get into trouble thousands of times more quickly than they can get out it. The consequences are: (1) poor and inconsistent user experience due to 'reliably unreliable' behaviour as networks saturate; (2) grossly over-provisioned and under-loaded networks to avoid saturation effects; and (3) fundamental mis-alignment of ISP and infrastructure cost and revenue models.

---

<sup>1</sup>  $O(n)$ ,  $n$  ratio of potential offered load from all ports on switch, over the rate the output port operates at. Typical values of 'n' in access networks range over 250 to 500 or more. The recovery rate is very low, a busy network link might only be able manage an (equivalent)  $n$  of 0.25 (slack) or less.

## Wrong assumptions lead to wrong network architectures

It has been a dominant meme that networks should be left to their own devices and that, when that is the case, the outcome (for each individual stream) is 'fair'. What is typically implied when networks are 'left to their own devices' is that there are first-in-first-out (FIFO) queues. The best that FIFO delivers is that it gives equal misery (i.e. fraction loss and same delay distribution) to each of the streams. Furthermore, this *only* occurs if the stream's arrival patterns are Poisson (have a negative exponential distribution between inter-arrival times).

Such arrival patterns do not occur in practice (for various reasons). What actually happens is the packets arrive bunched, they have phase-related interactions and therefore the interaction with the control loops (which are in TCP) give highly biased (and unfair) outcomes. The results are effectively non-deterministic. Such non-determinism results in features like the 'world-wide-wait' where your connection doesn't succeed until you abandon it and try again - it got into phase with some other connection and was always losing; stopping and trying again changes the relative phase of the two streams and hence the delivered loss and delay characteristics and it then succeeds. Network collapse due to 'buffer bloat' is another example of this phenomenon.

There is also another meme which is if you 'manage' the traffic everywhere (e.g. implement QoS as interpreted by your network equipment vendor) you can get an end-to-end service. This is neither necessary nor sufficient. The sufficiency point is related to phase issues, somewhat like above, that current traffic management approaches don't consider (or address).

## A better approach: pre-contending traffic at a single control point

The general Internet performance architecture is to start with low capacity connections, multiplex them into larger capacity connections up to a set of core cross-connected routers, and then multiplex them down into lower-capacity connections for delivery. Although the topology could be an arbitrary graph, routing protocols (and other constraints) mean that it is structured as a hierarchical tree around the edges, with core being more mesh-like. The main quality attenuation occurs at the points where 'fast' goes into 'slow'. Given topology information of the rest of the route (technology types and link speeds - either from supplier or empirically ascertainable) it is feasible to 'pre-contend' the traffic at the root of the hierarchical distribution point. This removes the need for *any* QoS mechanism in the other elements – as the emergent traffic stream has the appropriate gaps between packets so that no downstream contention for resources would occur.

This is a key point – as it brings all the issues into a *single location*, it means that other effects (such as traffic phase issues) can be resolved intelligently as well. In the long-term it provides for significant cost savings as all the down stream equipment can be dramatically simplified.

In the upstream path, the major fast-to-slow point is the egress from the premises, from that point onward a) things flow into fast pipes and b) those pipes are lightly used. This is because (around the edge) the services are delivered asymmetrically (downstream:upstream capacity in ratios of 20:1 not uncommon) and the usage is similarly asymmetric. This leads to extremely low peak utilization and very low quality attenuation.